Blue text on a black background

Description automatically generated

**המחלקה להנדסת חשמל ואלקטרוניקה**

**התקנים לוגיים מתוכנתים (31551)**

**פרויקט מסכם**

**Matrics Muliply**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

מנחה: ד"ר פאדל טריף

תאריך: 30/08/2024

תוכן ענינום

1. תיאור כללי...................................................................3
2. סכמת בלוקים כללית......................................................3
3. הגדרת הדרישות...........................................................4
4. ציוד נדרש....................................................................4
5. הנחיות כלליות..............................................................4
6. תכן מפורט...................................................................5
7. תיאור משימות..............................................................6
8. Matrices multiply....................................................6
9. Main controller.....................................................14
10. בדיקת המערכת בעזרת Signal TAP..............................25
11. סרטון הדגמה.............................................................29

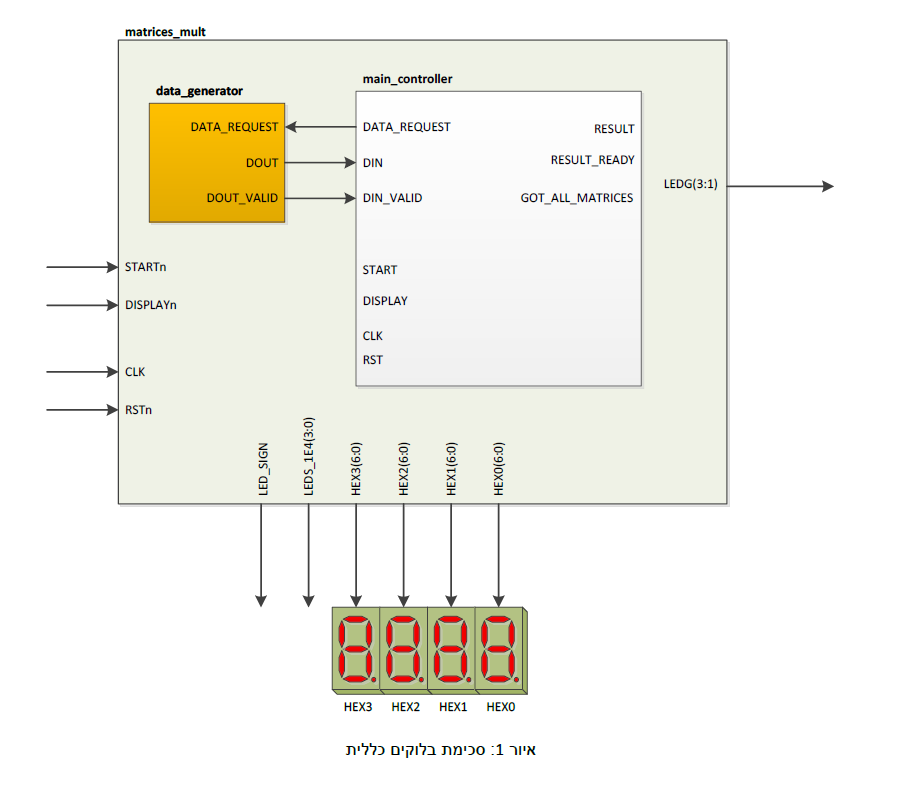
1. **תיאור כללי**

יש לתכנן מערכת שתקלוט שתי מטריצות בגודל 4×4 כל אחת, תחבר את תוצאת המכפלה שלהן, ותציג את התוצאה על גבי תצוגת 7-segment. המערכת תשתמש במעבד Cyclone V Starter Kit ותקבל קלטים מהמשתמש באמצעות כפתורי לחיצה. המערכת תתחיל במצב המתנה, ותפעל כאשר מתקבלות שתי המטריצות. בסיום חישוב המכפלה, התוצאה תוצג על גבי התצוגה.

1. **סכימת בלוקים כללית**

המערכת מורכבת ממספר בלוקים עיקריים:

1. **data\_generator:** אחראי על יצירת המטריצות לכניסה. בלוק זה יספק את המטריצות לחישוב.
2. **main\_controller:** הבלוק הראשי של המערכת שמנהל את כל התהליך – קבלת המטריצות, חישוב התוצאה, ורישום התוצאה בזיכרון.
3. **matrices\_mult:** אחראי על חישוב המכפלה בין שתי המטריצות.
4. **DISPLAY:** תציג את התוצאה על גבי תצוגת ה-7-segment.



**הבלוק הכתום יספק את המטריצות עליהן יבוצע החישוב. בלוק זה יסופק על ידינו ויהיה חלק מהקושחה הכוללת.**

**שימו לב: הבלוק main\_controller הוא הבלוק הראשי אך לא היחיד. כל השאר לא מופיעים באיור זה.**

1. **הגדרת הדרישות**

* **שעון המערכת:** המערכת תפעל על שעון יחיד של 50MHz.
* **RESET:** הכניסה היא active low, כלומר היא מופעלת כאשר הערך בכניסה הוא '0'.
* **מצב המתנה:** במצב המתנה כל ה-LEDs ותצוגת ה-7seg יהיו כבויים למעט LEDG(1) שיהיה דולק לאורך כל זמן פעולת המערכת.
* **קבלת מטריצות:** לחיצה ראשונה על לחצן STARTn תגרום ל-data\_generator להוציא את שתי המטריצות. סדר קבלת האיברים לכל מטריצה הוא משמאל לימין ומלמעלה למטה.
* **רישום בזיכרון:** המטריצות ירשמו לזיכרון בסדר שנקבע ע"י המשתמש.
* **חישוב מכפלה:** לאחר קבלת המטריצות, המערכת תבצע חישוב מכפלת המטריצות ותשמור את התוצאה בזיכרון.
* **מצב הצגה:** לאחר חישוב התוצאה, המערכת תעבור למצב הצגה. תוצג כל פעם ספרה אחת של התוצאה על גבי תצוגת ה-7-segment.
* **הצגה בלולאה:** לאחר הצגת כל האיברים של התוצאה, לחיצה נוספת על לחצן DISPLAYn תציג שוב את האיבר הראשון בלולאה.
* **התחלה מחדש:** לחיצה על הלחצן STARTn תחזיר את המערכת למצב המתנה.
* **יעילות ואופטימיזציה:** ינתן דגש על זמן החישוב של מטריצת היעד וכמות המכפלים בשימוש. יש למצוא את האיזון הנכון בין השניים.

1. **ציוד נדרש**

* **ערכת Cyclone V Starter Kit.**
* **מחשב הכולל תוכנות notepad++, modelsim ו-quartus.**

1. **הנחיות כלליות**

* **כתיבת הקוד:** הקוד יכתב ב-VHDL ויש לכלול testbench לכל בלוק.
* **סימולציה מלאה:** יש לבצע סימולציה מלאה לכל המערכת.
* **הגשת קוד אישי:** כל הקוד שיוגש חייב להיות אישי וללא שימוש בקוד מוכן ממקורות אחרים.
* **אופטימיזציה:** יש להקפיד על כתיבה מבנית, שימוש בסינכרונייזרים, ושמירה על כללי הכתיבה הנכונה לסינתיזה.
* **שמות קבועים:** חובה לשמור על שמות הקבצים, שמות ה-entities, שמות ה-ports וה-generics המוגדרים במסמך זה.

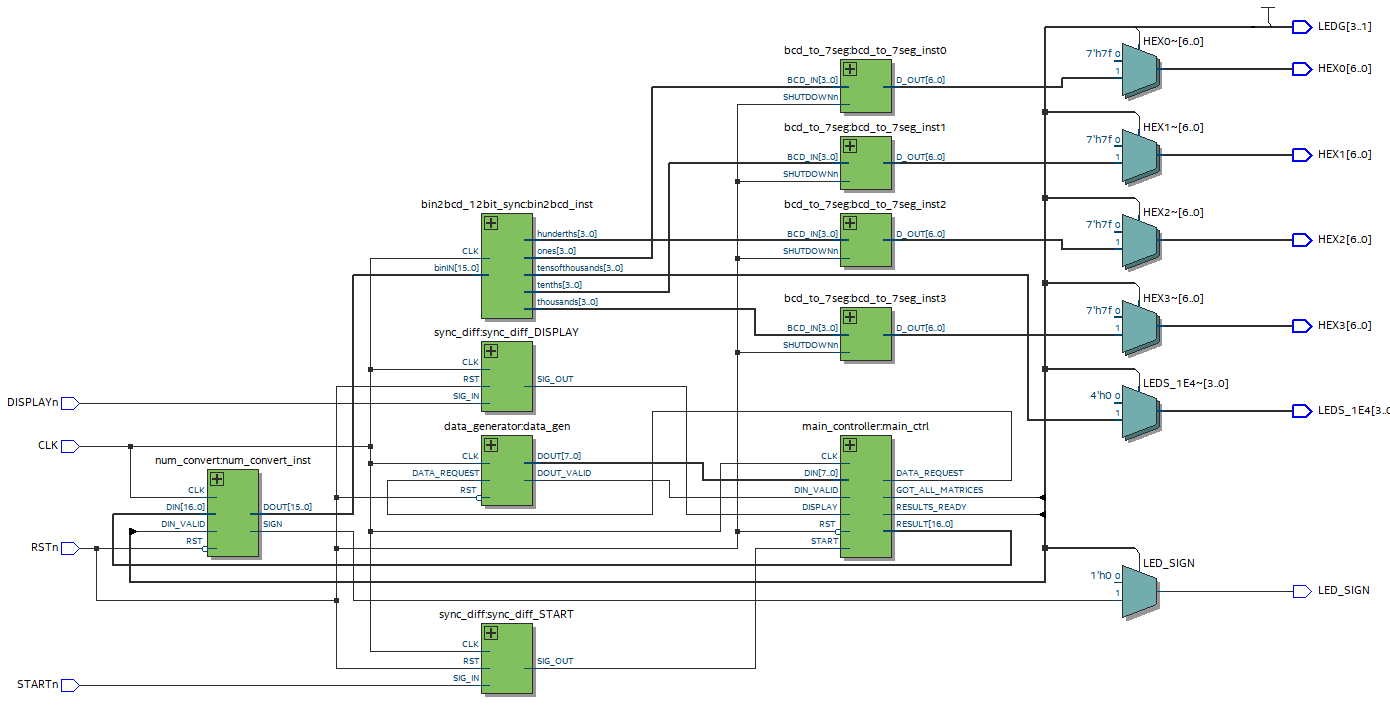
1. **תכן מפורט**
2. **תיאור הבלוק matrices\_mult:**

* זהו הבלוק העליון של המערכת, שמחבר בין כל הבלוקים האחרים למטרת הכפלת מטריצות והצגת התוצאות. הבלוק מקבל אותות שעון ואיפוס, ומנהל את זרימת הנתונים בין הבלוקים השונים כמו data\_generator, main\_controller, ו-num\_convert. כמו כן, הוא דואג להמרת התוצאה הבינארית לפורמט BCD ולהצגת התוצאה על גבי תצוגות 7-segment ו-LEDים. הבלוק הזה מתאם את כל התהליכים במערכת כדי להבטיח ביצועים נכונים ותוצאות מדויקות.

1. **תיאור הבלוק main\_controller:**
   * בלוק זה הוא הבלוק הראשי של המערכת שמנהל את כל התהליך – קבלת המטריצות, חישוב התוצאה, ורישום התוצאה בזיכרון.
2. **תיאור הבלוק data\_generator:**
   * בלוק זה אחראי על ייצור המטריצות לכניסה. הבלוק יוציא את המטריצות לפי דרישת המשתמש ויתעלם מפקודות בזמן הוצאת נתונים.
3. **תיאור הבלוק my\_multiplier:**
   * בלוק זה מכפיל שני מספרים (signed/unsigned) ומציג את התוצאה לאחר מספר מחזורי שעון שנקבע מראש.
4. **תיאור הבלוק bcd\_to\_7seg:**
   * הבלוק אחראי על המרת BCD לפורמט המתאים לתצוגת 7-segment.
5. **תיאור הבלוק bin2bcd\_12bit\_sync:**
   * בלוק זה ממיר מספר בינארי למספר BCD באורך של עד 12 ביטים.
6. **תיאור הבלוק matrix\_ram:**
   * בלוק זה אחראי על רישום וקריאה של המטריצות והתוצאה מהזיכרון.
7. **תיאור הבלוק num\_convert:**
   * הבלוק אחראי על המרת מספרים לא חתומים (unsigned) למספרים חתומים (signed) ולהיפך.
8. **תיאור הבלוק sync\_diff:**
   * בלוק זה אחראי על סנכרון אותות אסינכרוניים עם השעון, על מנת למנוע בעיות של metastability במערכת. הבלוק מבטיח שהאות הנכנס יעבור סינכרון ויהיה זמין בצורה יציבה לשימוש בשאר חלקי המערכת.
9. **תיאור משימות**

**משימה 1: מימוש הרמה העליונה matrices\_mult**

* + 1. **תכנון הבלוק matrices\_mult**

****

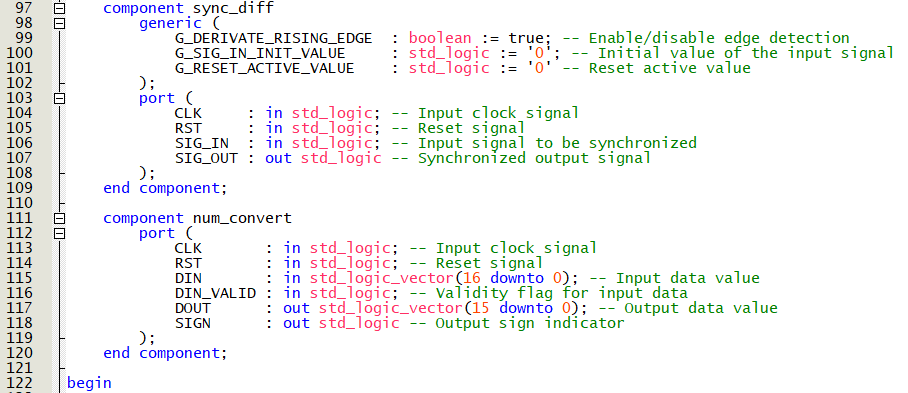
A screenshot of a computer

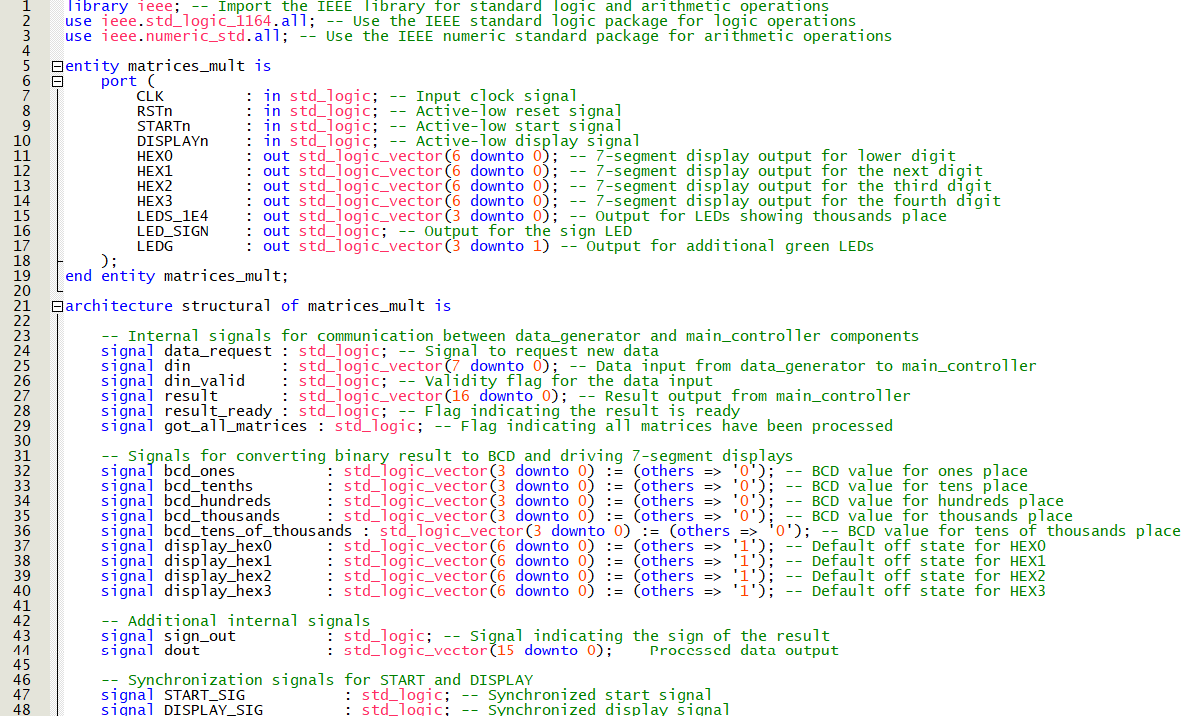
Description automatically generated

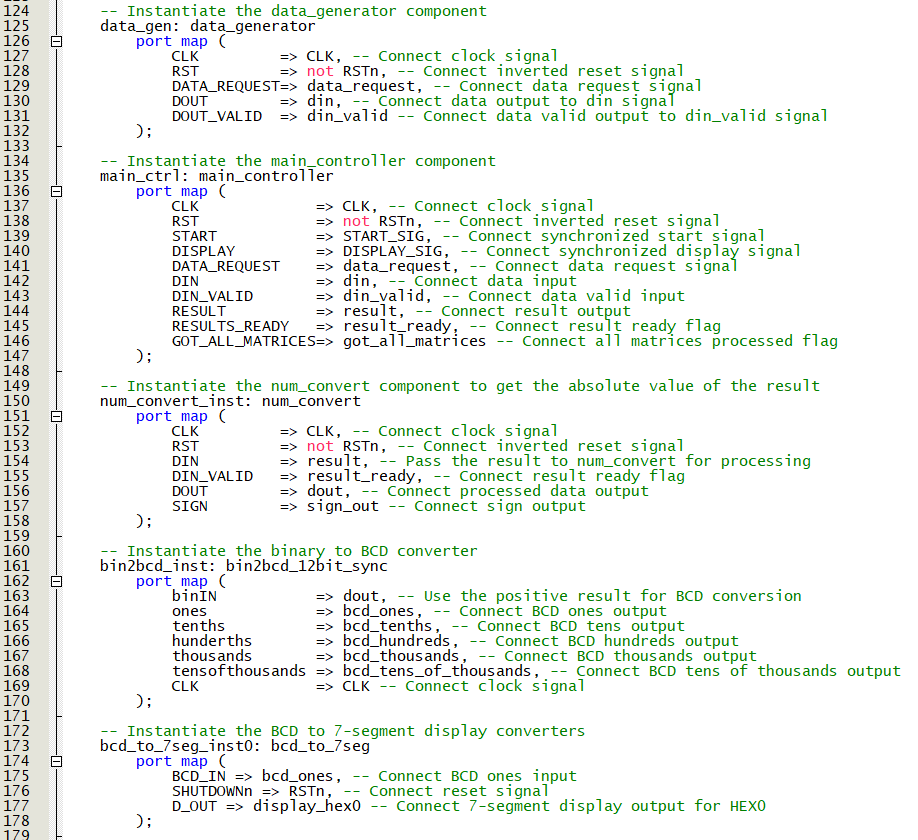
**מבנה כללי:**

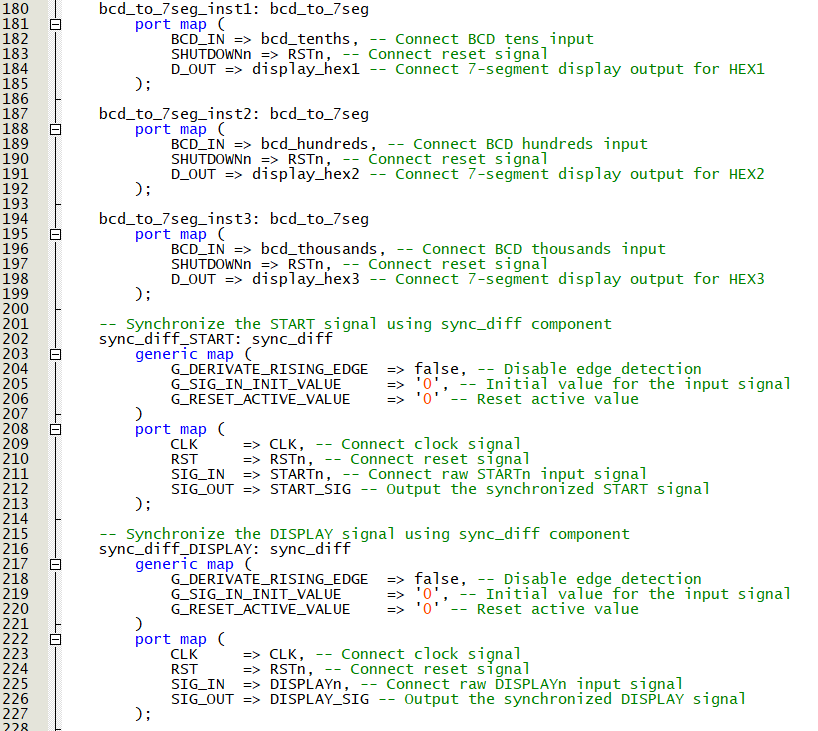
היישות matrices\_mult מקבלת מספר אותות כניסה, כולל שעון (**CLK**), אות איפוס (**RSTn**), אות התחלה (**STARTn**), ואות תצוגה (**DISPLAYn**). היא מבצעת את הכפל של שתי מטריצות ומציגה את התוצאה באמצעות תצוגות **7-segment** ונורות **LED**.

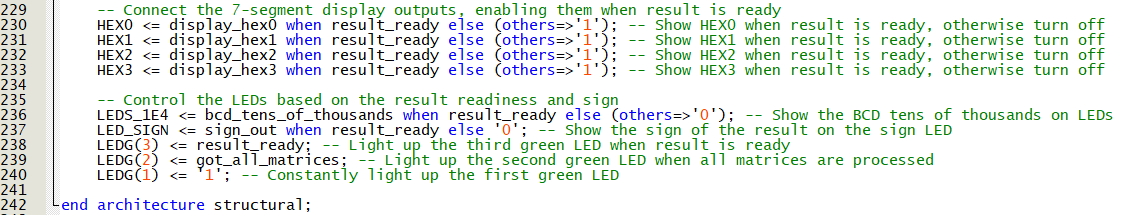
**2.1.2 סימולציה פונקציונאלית של הבלוק matrices\_mult**

* ****A screenshot of a computer program

  Description automatically generated**matrices\_mult code:**

****

****

****

* **אותות ומסרים פנימיים:**

1. **data\_request, din, din\_valid:** משמשים לתקשורת בין היישות המרכזית (**main\_controller**) ליחידת ייצור הנתונים (**data\_generator**). היישות המרכזית מבקשת נתונים, מקבלת אותם, ומסמנת האם הם תקפים.
2. **result, result\_ready, got\_all\_matrices:** אותות שמעידים על קבלת התוצאה, מוכנות התוצאה וסיום תהליך קבלת המטריצות בהתאמה.

* **הסבר על רכיבי המערכת:**

1. **data\_generator:** אחראי על יצירת הנתונים לשימוש ביישות המרכזית. הוא יוצר את המטריצות שצריך להכפיל.
2. **main\_controller**: היישות המרכזית שמנהלת את כל תהליך הכפל. היא מקבלת את הנתונים מהמטריצות, מבצעת את החישובים, ושומרת את התוצאה.
3. **num\_convert:** ממיר את התוצאה הבינארית לפורמט שבו ניתן להציג על תצוגת 7-segment, כולל קבלת הערך המוחלט במקרה של מספרים שליליים.
4. **bin2bcd\_12bit\_sync:** ממיר את התוצאה הבינארית לפורמט **BCD** כדי שניתן יהיה להציג אותה על תצוגות **7-segment**.
5. **bcd\_to\_7seg**: ממיר כל ספרת **BCD** לקוד תצוגת **7-segment** המתאים.
6. **sync\_diff**: רכיב שמשתמש לסנכרון האותות האסינכרוניים כמו **START** ו-**DISPLAY**, על מנת למנוע בעיות של אי יציבות במערכת (metastability).

* **תהליך הפעולה:**

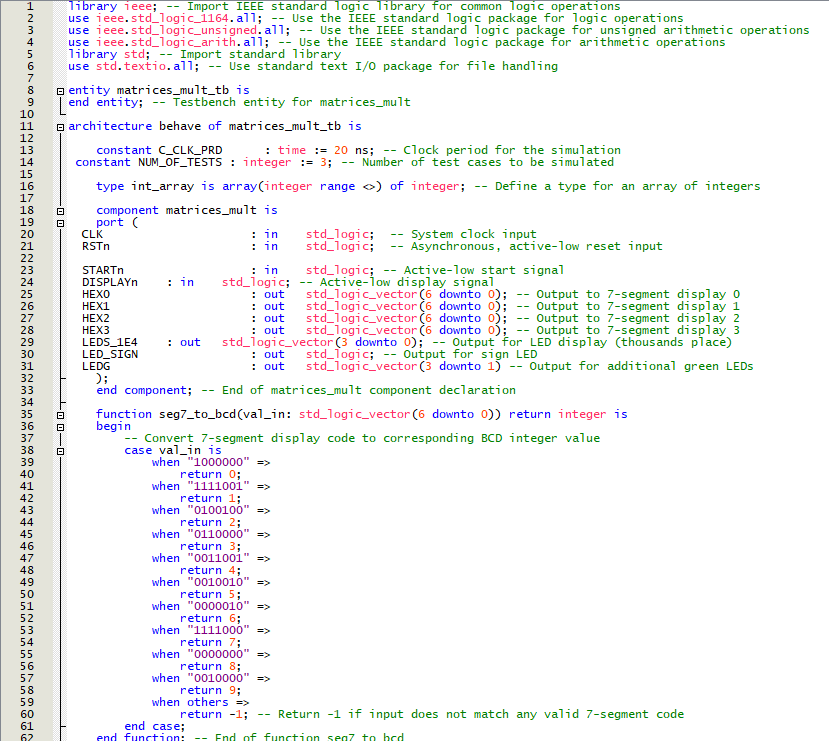
1. **סנכרון אותות:** האותות **STARTn** ו-**DISPLAYn** מסונכרנים באמצעות הרכיב **sync\_diff** כדי להבטיח שהמערכת תקבל את האותות בצורה יציבה.
2. **יצירת נתונים:** היישות המרכזית שולחת בקשה ליחידת ייצור הנתונים (**data\_generato**r) כדי לקבל את המטריצות שהיא צריכה להכפיל.
3. **כפל מטריצות:** היישות המרכזית (**main\_controller**) מבצעת את הכפל של המטריצות שהתקבלו ושומרת את התוצאה.
4. **המרת תוצאה לפורמט תצוגה:** לאחר קבלת התוצאה, היא מומרת לפורמט **BCD** באמצעות **bin2bcd\_12bit\_sync,** ולאחר מכן מומרת לקוד תצוגת **7-segment** באמצעות **bcd\_to\_7seg**.
5. **הצגת התוצאה:** התוצאה מוצגת על ארבע תצוגות **7-segment** ועל נורות **LED** שמציגות את הספרה הראשונה (אלפי) ואת סימן המספר (חיובי או שלילי).

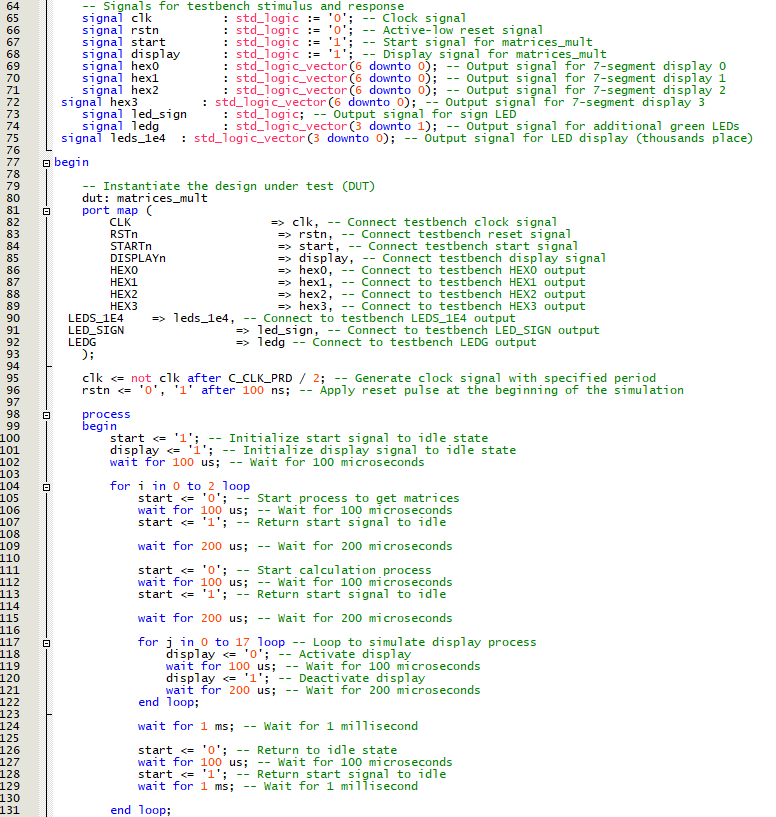
* **שליטה בתצוגה ובנורות LED:**

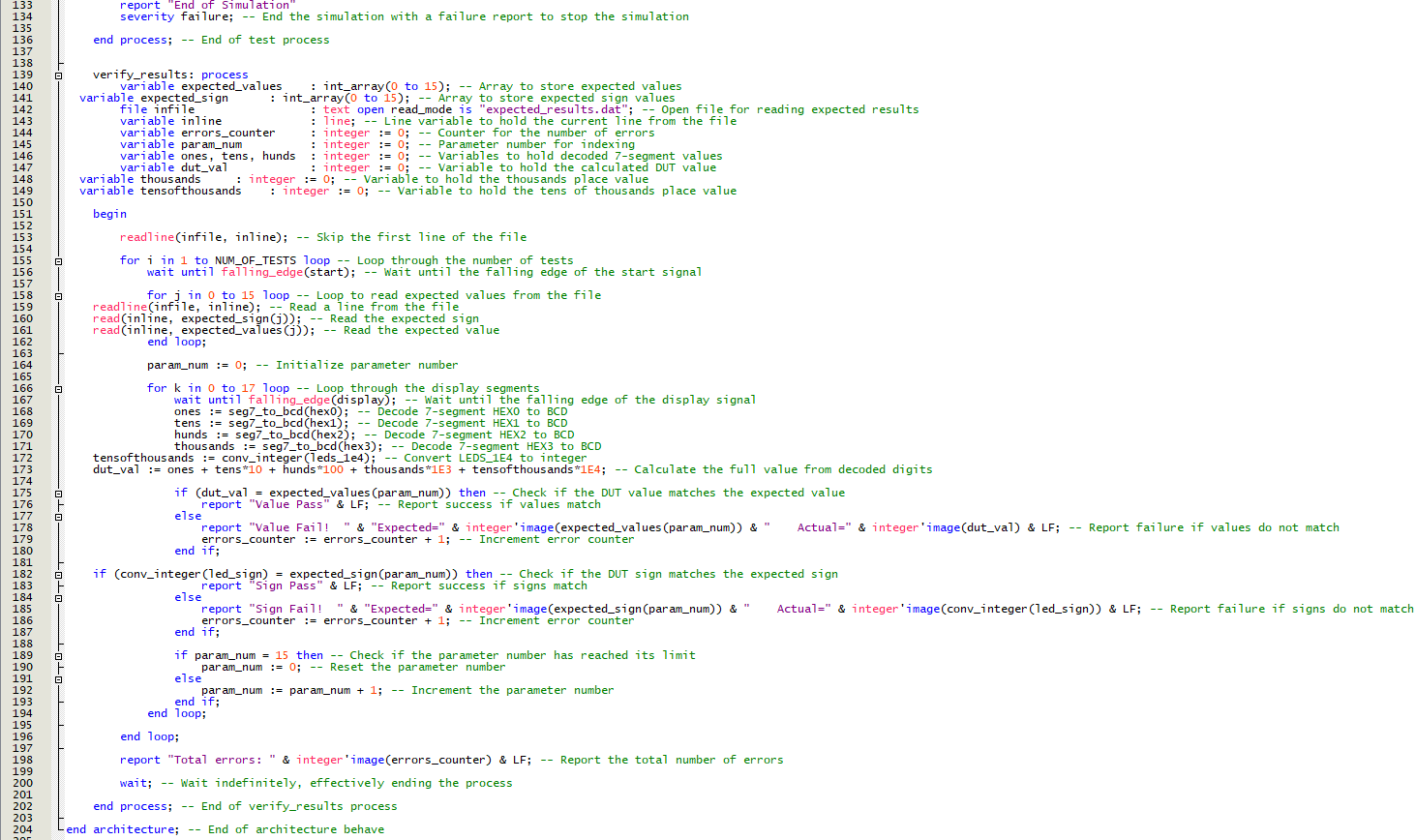
1. **HEX0-HEX3:** מציגים את הספרות השונות של התוצאה על תצוגות **7-segment**.
2. **LEDS\_1E4:** מציג את הספרה במיקום האלפי.
3. **LED\_SIGN:** מציג את הסימן של התוצאה (חיובי או שלילי).
4. **LEDG:** נורות ירוקות שמסמנות שהמערכת מוכנה עם התוצאה, ושהמטריצות התקבלו.

* **סיכום:**

הקוד הזה מגדיר מערכת מורכבת שמשלבת כמה רכיבים לשם ביצוע כפל מטריצות והצגת התוצאה בצורה ברורה. הוא משתמש במכונת מצבים ובאותות סנכרון כדי לוודא שכל שלב בתהליך מתבצע בצורה תקינה ולוגית.

* **matrices\_mult\_TB:**

****

****

* A computer screen shot of a program

  Description automatically generated**matrices\_mult simulation:**

ر

ر

ر

**הסבר סימולציה:**

**קבלת מטריצות והתחלת חישוב:**

* תחילת מצב: כאשר האות STARTn יורד לאפס.
* לחיצה ראשונה על START ,המערכת מתחילה לקבל את נתוני המטריצות דרך אות ה-din. כל האותות המעידים על קבלת הנתונים (=1data\_request, din\_valid) פעילים ומסמנים שהנתונים נקלטו אחרי ש got\_all matrices עלה ל1.
* לחיצה שניה על START , מתחילה חישוב המכפלה בין המטריצות. תהליך זה נמשך עד לסיום החישוב ואז רואים ש result\_ready עולה ל1.

**הצגת התוצאה:**

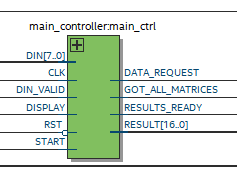
* תחילת מצב: כאשר האות DISPLAYn מופעל.
* לחיצה על DISPLAYnמציגה את האברים , התוצאה הסופית מוצגת על תצוגות ה-7-segment (HEX0-HEX3), כאשר אותות ה-LED מסמנים את מצב התוצאה (חיובי/שלילי). המערכת ממשיכה במצב זה עד לקבלת פקודת איפוס חדשה.

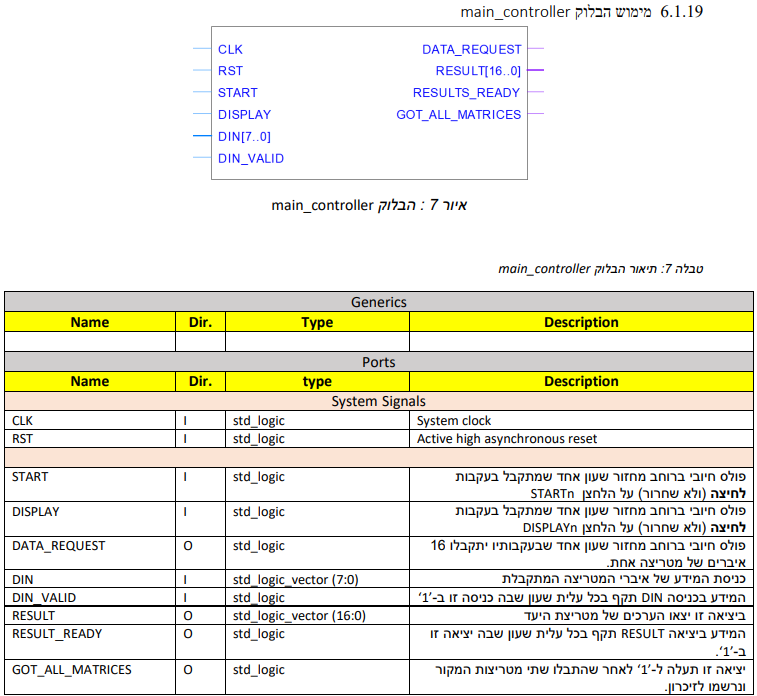
**לחיצת בפעם שלישית לSTART :**

* לחיצה שלישית על START , המערכת מתחילה לקבל את נתוני המטריצות
* השניות ועוד שתי לחיצות מתילים לראות תוצאות המטריצה השניה.

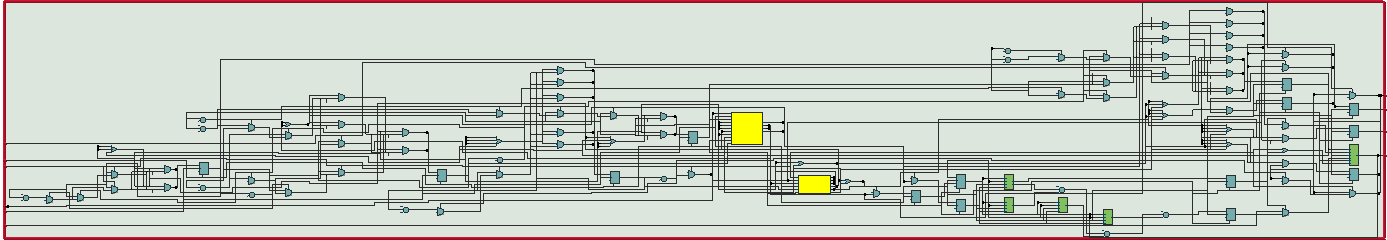
**START\_SIG DISPLAY\_SIG :**

* אלה הsignals של SRART וDISPLAY אחרי שהכנסנו אותם ל SYNC כדי שיהיו באורך שעון אחד בלבד והן סנכרונים

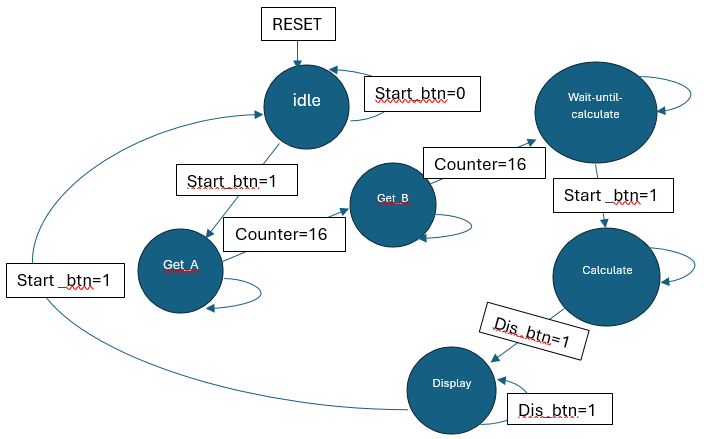
**משימה 2: ייצור הבלוק main\_controller**

* 1. **הבלוק main\_controller**

**מכונת מצבים**

****

**מכונות מצבים:**

1. **Main state Machine**

**תיאור המצבים במכונת המצבים הראשית (main\_sm):**

1. **st\_idle:**

במצב זה, המערכת מחכה לאתחול (באמצעות אות START). כאשר מתקבל האות, המערכת עוברת למצב **st\_receive\_first\_mat** שבו היא מתחילה לקבל את המטריצה הראשונה.

1. **st\_receive\_first\_mat:**

המערכת נמצאת במצב קבלת המטריצה הראשונה. היא מקבלת נתונים דרך האות **DIN** ומאמתת אותם באמצעות **DIN\_VALID**. לאחר קבלת כל הנתונים למטריצה הראשונה, המערכת עוברת למצב קבלת המטריצה השנייה **st\_receive\_second\_mat**.

1. **st\_receive\_second\_mat:**

במצב זה, המערכת מקבלת את המטריצה השנייה. תהליך זה דומה לתהליך של קבלת המטריצה הראשונה. לאחר קבלת כל הנתונים, המערכת עוברת למצב המתנה לחישוב **st\_wait\_for\_calculate.**

1. **st\_wait\_for\_calculate:**

במצב זה, המערכת מחכה לקבלת אישור נוסף (אות **START**) כדי להתחיל את תהליך החישוב. כאשר מתקבל האות, המערכת עוברת למצב **st\_calculate.**

1. **st\_calculate:**

זהו מצב שבו מתבצע החישוב של כפל המטריצות. כאן מכונת המצבים המשנית (**calc\_sm**) נכנסת לפעולה ומבצעת את החישוב עבור כל שורה ועמודה במטריצות.

1. **st\_display:**

לאחר שהחישוב הושלם, המערכת עוברת למצב התצוגה שבו התוצאה מוצגת. כאשר מתקבל האות **DISPLAY**, המערכת מציגה את התוצאה על המסך או תצוגה אחרת. לאחר הצגה של כל התוצאה, המערכת חוזרת למצב **st\_idle**.

1. **A diagram of a number system

   Description automatically generatedMultiply State Machine**

**תיאור המצבים במכונת המצבים של החישוב (calc\_sm):**

**st\_idle:**

מצב מנוחה שבו מחכה המערכת להתחלת תהליך החישוב.

**st\_get\_row:**

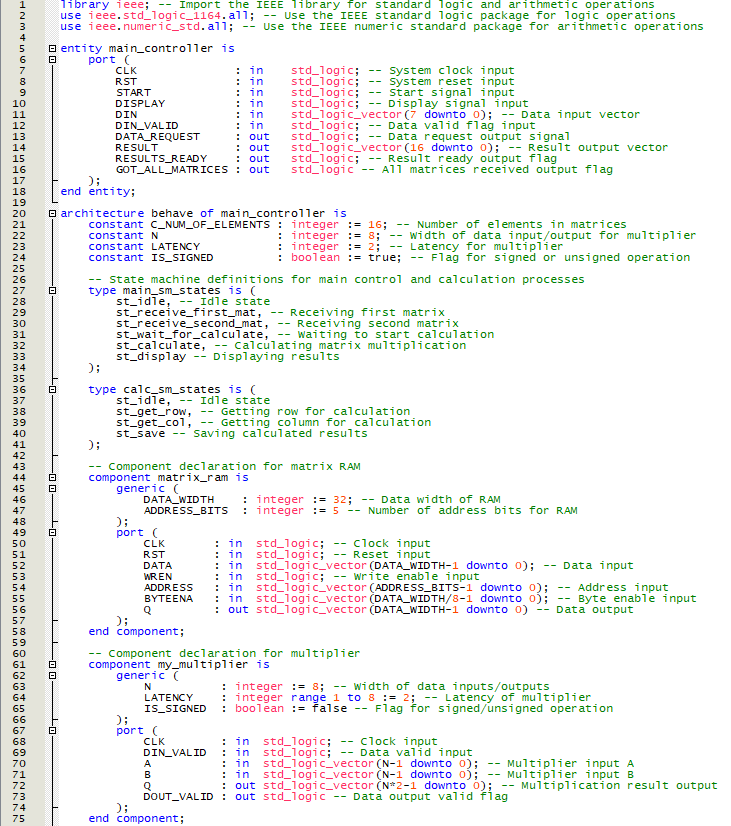
במצב זה, המערכת משיגה את השורה מהמטריצה הראשונה לצורך חישוב.

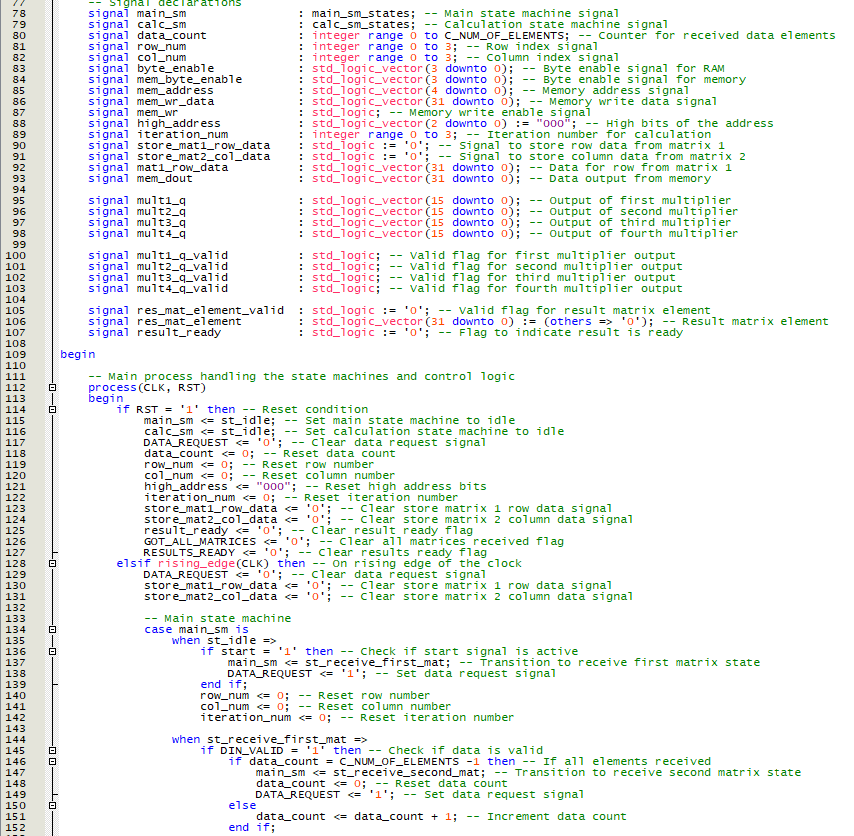
**st\_get\_col:**

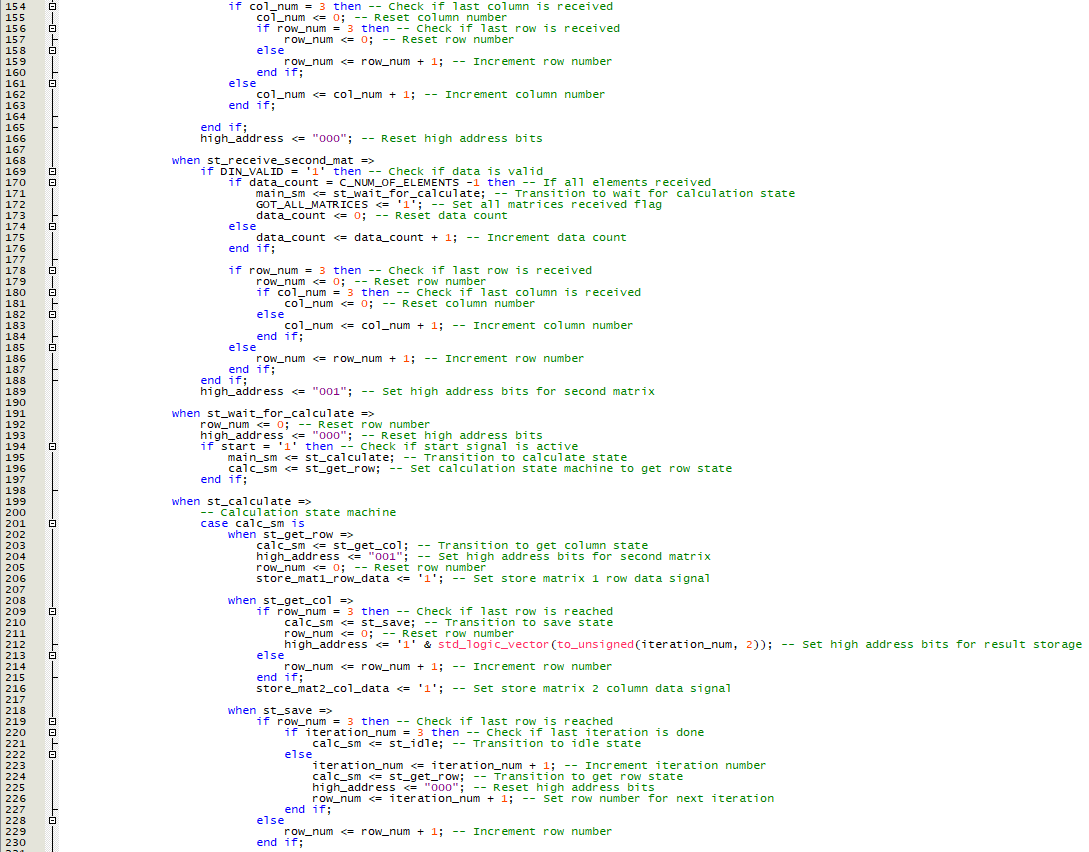
במצב זה, המערכת משיגה את העמודה מהמטריצה השנייה לצורך חישוב כפל של השורה והעמודה. לאחר מכן, היא עוברת למצב **st\_save**.

**st\_save:**

במצב זה, המערכת שומרת את התוצאה של כפל השורה והעמודה במשתנה תוצאה. אם כל השורות והעמודות עובדו, המערכת חוזרת למצב **st\_idle**.

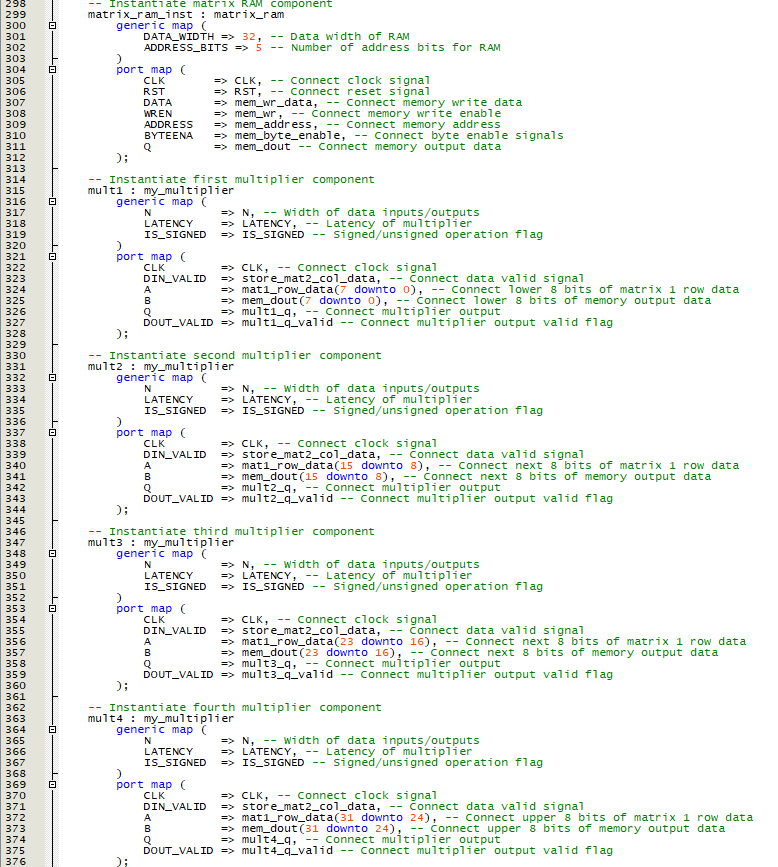
* **** **main\_controller code:**

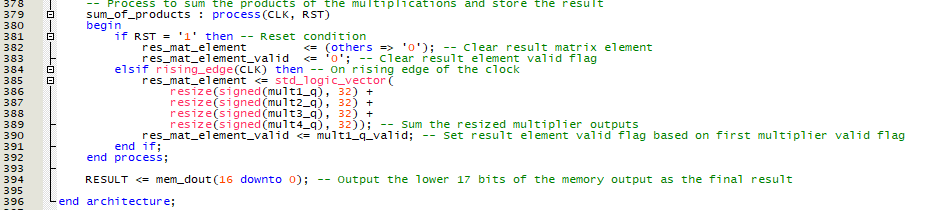
****

****

**A screenshot of a computer code

Description automatically generated**

****

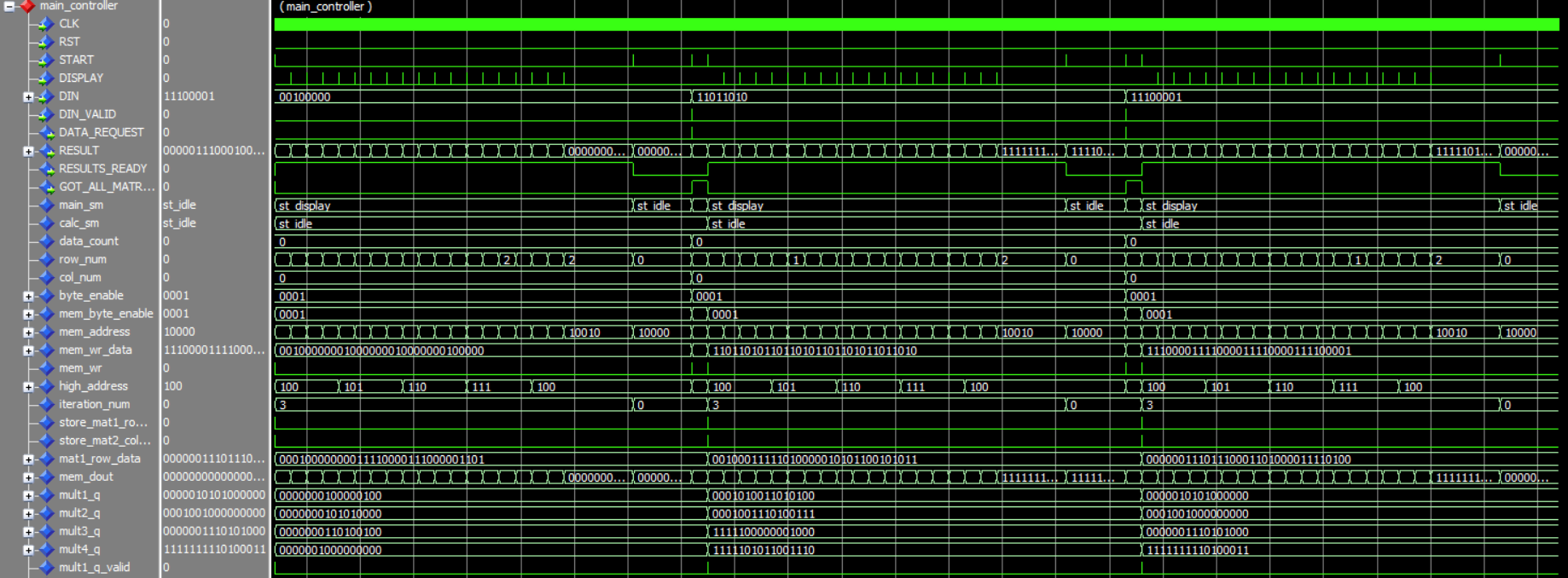
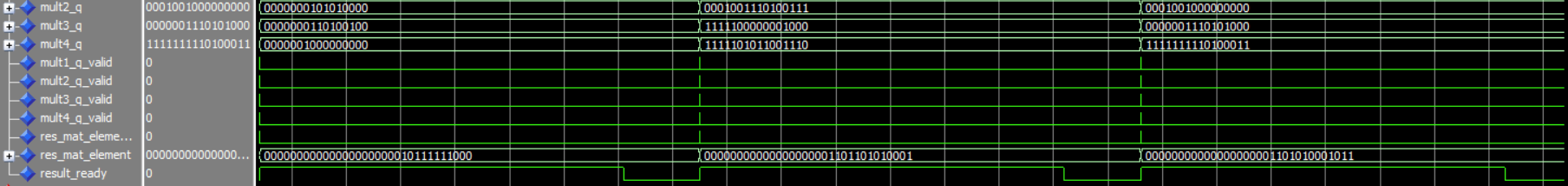
****

**הסברים כלליים:**

* **הגדרות ראשוניות:** הקוד כולל קבועים C\_NUM\_OF\_ELEMENTS, N, LATENCY)) המשמשים להגדרת מאפיינים שונים של המערכת, כמו מספר האיברים במטריצה, רוחב הנתונים, וזמן ההשהיה של הכפל.
* **מכונת מצבים** (**State Machine**)**:** הקוד משתמש בשתי מכונות מצבים - אחת לתהליך הראשי (**main\_sm**) ואחת לתהליך החישוב. (**calc\_sm**) כל אחת מהן עוברת בין מצבים שונים כדי לבצע את המשימות הנדרשות.

**סיכום כללי:**

הקוד הזה מגדיר יחידה מרכזית שתפקידה לקבל נתונים של מטריצות, לבצע כפל מטריצות בצורה סינכרונית באמצעות מכפלים, ולשמור את התוצאה בזיכרון לשימוש בהמשך או להצגה. מכונות המצבים מבטיחות שהמעבר בין השלבים יתבצע בצורה נכונה ורציפה, תוך שמירה על הסדר הלוגי של הפעולות.

* **main\_controller simulation:**

**הסבר סימולציה:**

**קבלת המטריצות:**

* **תחילת מצב:**  כאשר האות **STARTn** עולה לגבוה.
* **לחיצה ראשונה על STARTn :** המערכת מתחילה לקבל את נתוני המטריצה הראשונה דרך אות ה**DIN** -. כל האותות המעידים על קבלת הנתונים (**DATA\_REQUEST, DIN\_VALID**) פעילים ומסמנים שהנתונים נקלטו כראוי. האות **GOT\_ALL\_MATRICES** עולה ל-1 כדי להראות שכל המטריצות התקבלו.
* **לחיצה שניה על STARTn :** המערכת מתחילה לקבל את נתוני המטריצה השנייה באותו תהליך כמו בקבלת המטריצה הראשונה. כאשר הנתונים התקבלו, המערכת מוכנה להתחיל את תהליך החישוב.

**חישוב המטריצות:**

* **תחילת מצב**: כאשר האות **STARTn** מופעל פעם נוספת לאחר קבלת שתי המטריצות.
* **תהליך:** המערכת מבצעת את חישוב המכפלה בין המטריצות. תהליך זה נמשך עד לסיום החישוב, שאז ניתן לראות שהאות **RESULTS\_READY** עולה ל-1 כדי לסמן שהתוצאה מוכנה.

**הצגת התוצאה:**

* **תחילת מצב:** כאשר האות **DISPLAYn** מופעל.
* **לחיצה על DISPLAYn :** התוצאה הסופית מוצגת על תצוגות ה (-segment HEX0-HEX3 7 ) כאשר האותות **LEDG** ו- **LED\_SIGN** מסמנים את מצב התוצאה (חיובי/שלילי). המערכת ממשיכה במצב הצגה זה עד לקבלת פקודת איפוס של **STARTn** חדשה.

**לחיצה נוספת על STARTn:**

* **תחילת מצב:** לחיצה שלישית על **START** מתחילה את התהליך מחדש, המערכת שוב מוכנה לקבל נתוני מטריצה חדשה ולהציג את התוצאות במידה והתקבלו שתי מטריצות חדשות.

**קבלת נתוני המטריצות והתחלת החישוב:**

* **תחילת מצב**: כאשר האותות **mult1\_q\_valid** עד **mult4\_q\_valid**  הופכים לגבוהים.
* **תהליך:** המערכת מקבלת את תוצאות החישובים הבינאריים מהחישובים של כל שורה ועמודה במטריצות (דרך **mult1\_q** עד**mult4\_q** ). כל אחד מהאותות האלה מסמן שהערך התקבל ומאומת. ברגע שהאות **res\_mat\_elements\_valid** עולה ל-1, המערכת מסיימת את קבלת הנתונים ועוברת לשלב הבא.

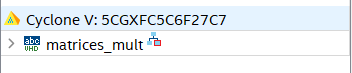
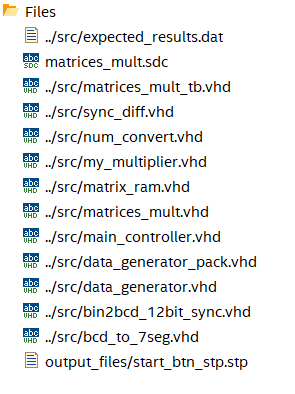
**חישוב תוצאת המטריצה:**

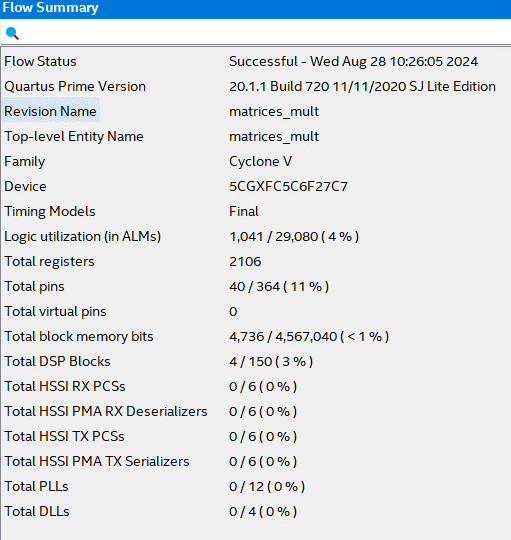
* **תחילת מצב:** כאשר האות **res\_mat\_elements\_valid**  במצב גבוה.
* **תהליך:** המערכת מתחילה לחשב את התוצאה של כפל המטריצות, כאשר האות **res\_mat\_elements\_q**  מציג את התוצאה של כל אלמנט במטריצה. תהליך זה ממשיך עד לסיום חישוב כל האלמנטים של המטריצה.

**סיום החישוב והצגת התוצאה:**

* **תחילת מצב:** כאשר האות **result\_ready**  עולה ל-1.
* **תהליך:** בסיום החישוב, האות **result\_ready** עולה כדי לסמן שהתוצאה הסופית מוכנה להצגה או לשימוש נוסף במערכת. בשלב זה, המערכת סיימה את החישוב ומוכנה לשלב הבא של הצגת התוצאה או קבלת פקודת איפוס חדשה.

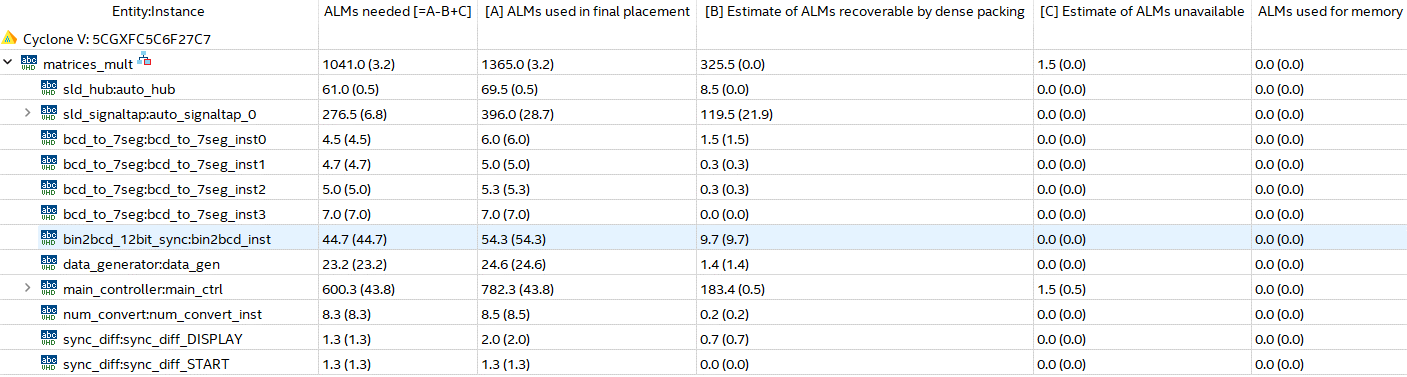
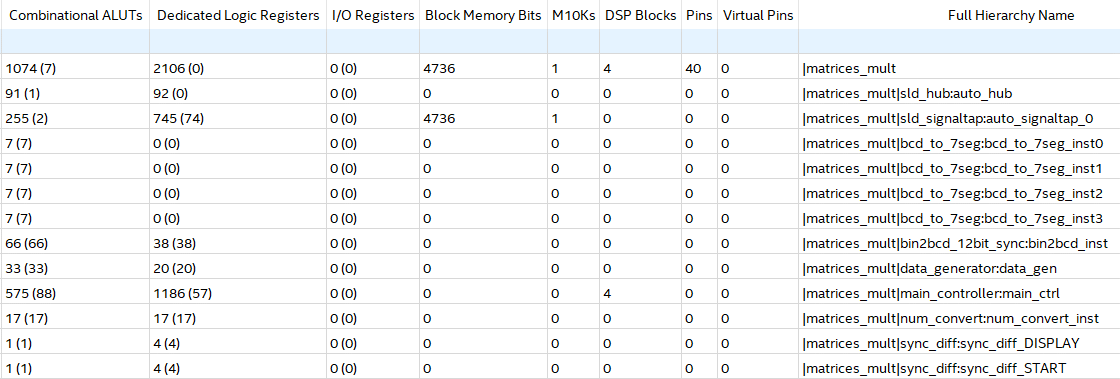
**3. שלבי הסינתיזה**

******3.1 הרמה העליונה והקבצים**

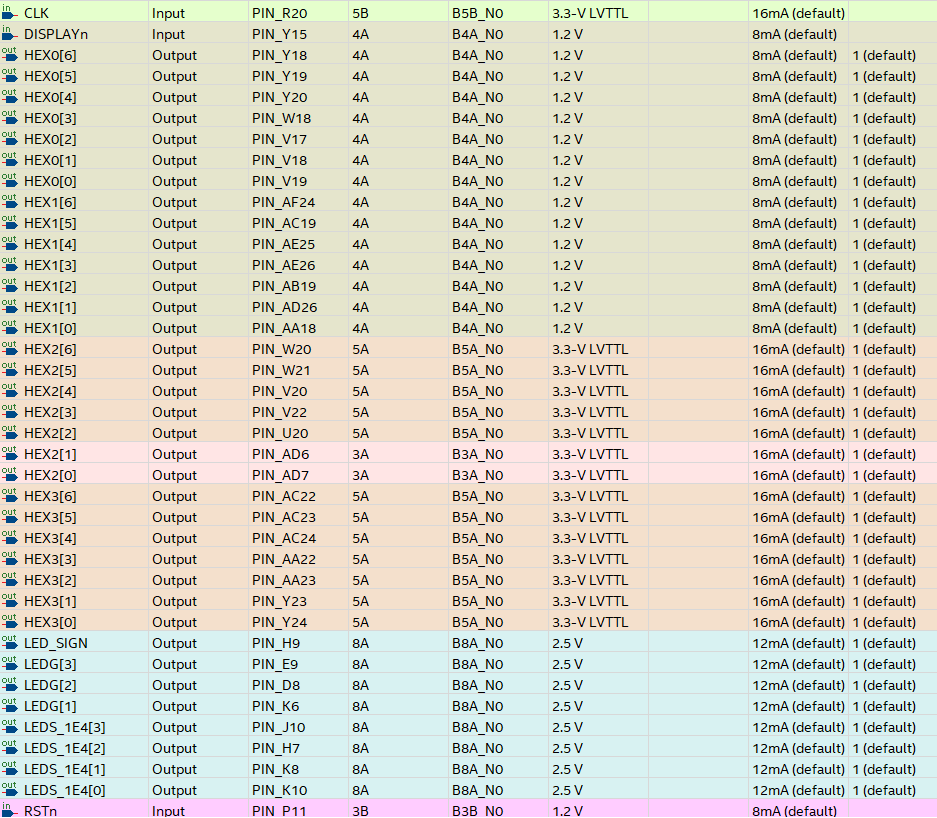
**3.2 קומפילציה (ניצול משאבים)**

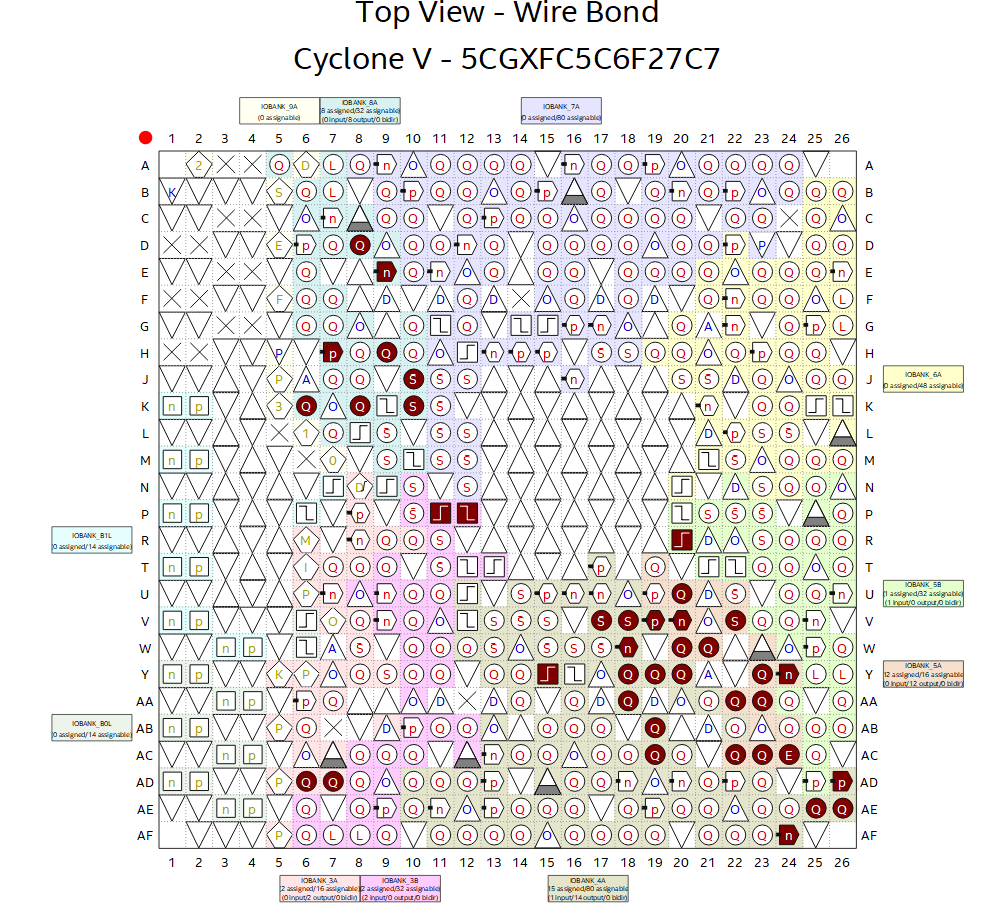
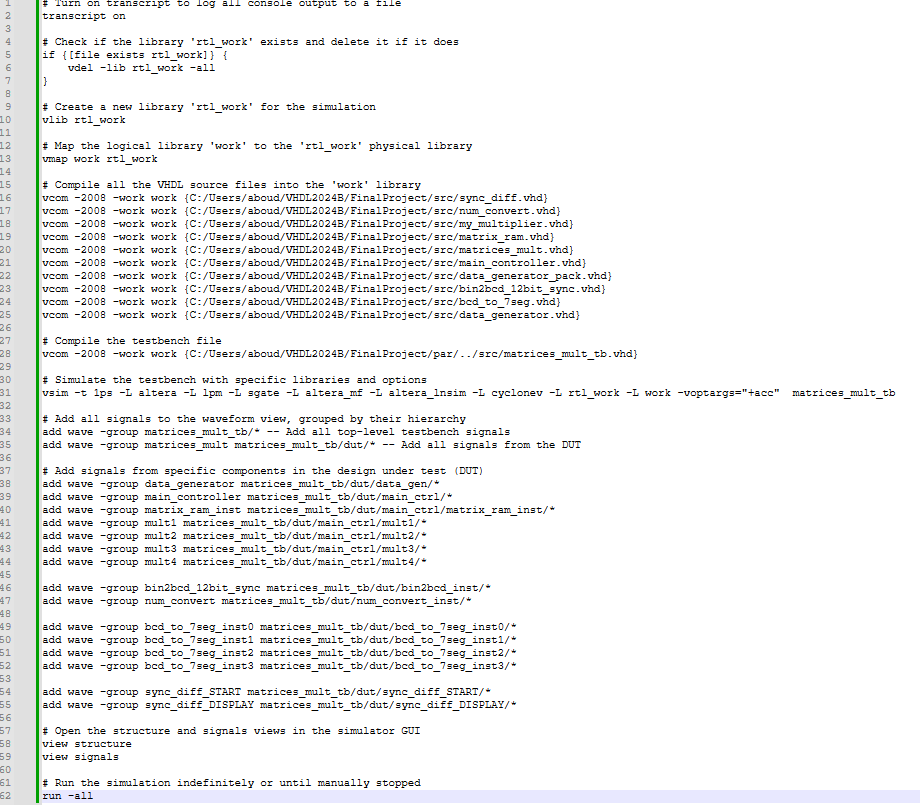
**הסבר למה השתמשנו ביותר DSP בלקים:**

בגלל שעשינו Signal tap היא צורכת יותר משאבים.

**3.3 ניצול משאבים של כל בלוק**

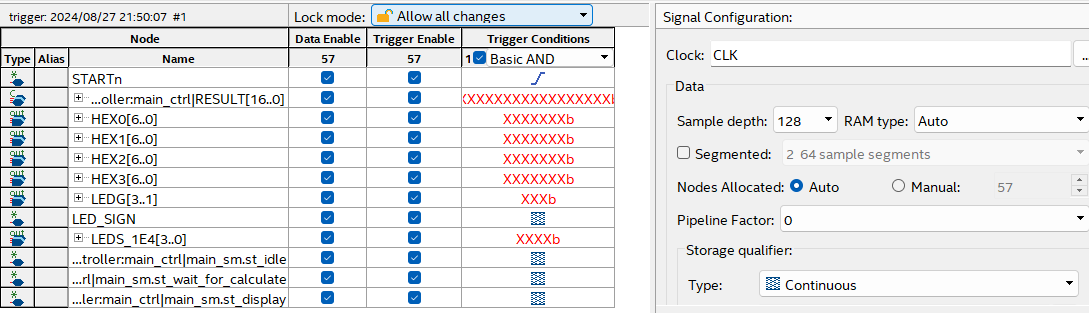
**3.4 הקצאת פינים.**

הקצאת הפינים נעשתה דרך קובץ שהורדנו מהמודל והוספנו אותו לפרויקט

****

**קובץ Do:**

1. **בדיקת המערכת בעזרת Signal TAP**

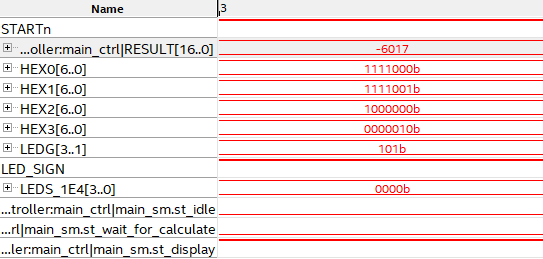
* **יצירת קובץ Signal TAP לבדיקת המערכת ללחיצת STARTn**
* **סימולציה דרך ה Signal TAP ללחיצת STARTn**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**הסבר איטרציות (העברה ממטריציה A ל B):**

* **בלחיצה STARTn 1:**

**אנחנו נמצאים במטריציה A, התוצאה 250:**

הערך 250 הוא תוצאת החישוב הראשון במטריציה A שמערכת הפיקה, והוא זה שאמור להיות מוצג על תצוגות ה-7-segment.

**תצוגות ה-7-segment (HEX0 עד HEX3):**

**כל HEX מייצג תצוגת 7-segment אחת:**

**HEX0**: מראה 1000000b שזהו הקוד להצגת הספרה 0 על תצוגת 7-segment.

**HEX1**: מראה 0100000b שזהו הקוד להצגת הספרה 2.

**HEX2**: מראה 0010010b שזהו הקוד להצגת הספרה 5.

**HEX3**: מראה 1000000b שזהו הקוד להצגת הספרה 0.

**נורות ה-LED (LEDG ו-LED\_SIGN):**

**[3:1] LEDG מציגים את מצב ה-LED הירוק:** 101b מציין שה-LED הראשון והשלישי דולקים והשני כבוי.

**LED\_SIGN:** ערך 0 מציין שהתוצאה חיובית (ללא סימן מינוס).

**מצבי מכונת המצבים:**

מכונת המצבים נמצאת במעבר בין המצבים **st\_idle, st\_wait\_for\_calculate**, ו-**st\_display**, מה שמראה שהמערכת בהתחלה מחכה לפעולה (**idle**), ואז ממתינה להתחלת החישוב (**wait\_for\_calculate**), ולאחר מכן מציגה את התוצאה (**display**).

* **בלחיצה STARTn 2 ו- 3:**

**כל HEX מייצג תצוגת 7-segment אחת שהי 1111111b שזהו הקוד לכבוי התצוגה**

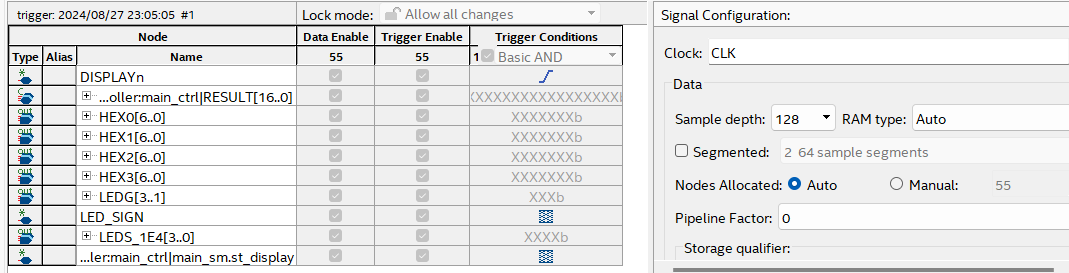
מכונת המצבים עברה למצב **idle** ואחר כך למצב **wait\_for\_calculate**

* **בלחיצה STARTn 4:**

**אנחנו עברנו ל****מטריציה B, התוצאה** **6017-:**

הערך 6017- הוא תוצאת החישוב הראשון במטריציה B שמערכת הפיקה.

**LED\_SIGN:** ערך 1 מציין שהתוצאה שלילי (עם סימן מינוס).

* **יצירת קובץ Signal TAP לבדיקת המערכת ללחיצת DISPALYn**
* **A screenshot of a computer

  Description automatically generatedA screenshot of a computer

  Description automatically generatedA screenshot of a computer

  Description automatically generatedA number in a row

  Description automatically generatedסימולציה דרך ה Signal TAP ללחיצת DISPLAYn**

A screenshot of a computer

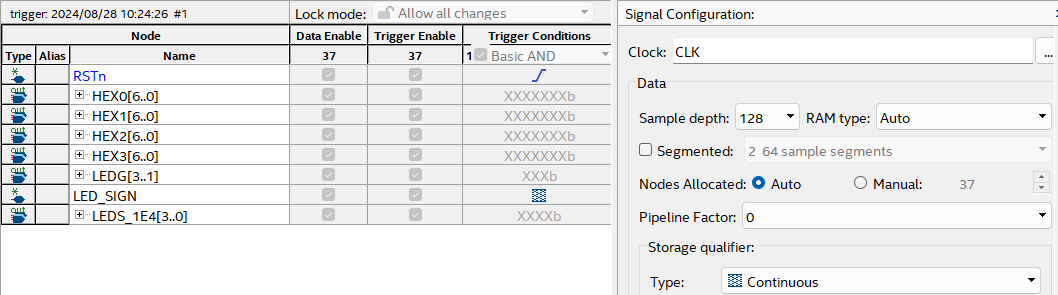
Description automatically generated

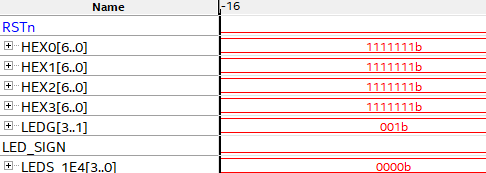
**הסבר איטרציות (העברה במטריציה A):**

* **בלחיצה DISPLAYn:**

**התוצאה עברה לאיבר השני 260**

**ובכל לחיצה על DISPLAYn התוצאה תעבור להאיבר הבא**

* **יצירת קובץ Signal TAP לבדיקת המערכת ללחיצת RSTn**

**סימולציה דרך ה Signal TAP ללחיצת RSTn**

* **בלחיצה RSTn:**

**כל HEX מייצג תצוגת 7-segment אחת שהי 1111111b שזהו הקוד לכבוי התצוגה**

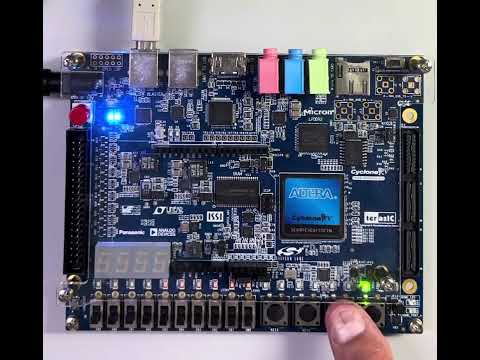
**ובכל לחיצה על DISPLAYn התוצאה תעבור להאיבר הבא**

**[3:1] LEDG מציגים את מצב ה-LED הירוק:** 000b מציין שכל ה-LEDs כבוים.

**LED\_SIGN:** ערך 0 מציין שה-LED כבויה.

1. **סרטון הדגמה:**

[**https://www.youtube.com/watch?v=htl3920-ROM**](https://www.youtube.com/watch?v=htl3920-ROM)

**[](https://www.youtube.com/embed/htl3920-ROM?feature=oembed)**